

Analiza cerintelor

Se cere implementarea unui program bazat pe MPI care sa faca suma a doua numere mai mari de 10 cifre. Cifrele numerelor se gasesc in fisiere, avand cifra cea mai nesemnificativa prima. Aceasta problema trebuie abordata in 3 variante: secvential, folosind MPI_Send si MPI_Recv, folosind MPI_Scatter.

Proiectare

Varianta 0 - secvential

Numerele sunt citite din fisier si stocate in doua tablouri. Cifra cea mai nesemnificativa este prima. Se aduna cifra cu cifra, daca apare carry, acesta este utilizat la adunarea urmatoarelor cifre. In final, daca exista cary, se adauga pe ultima pozitie 1. Rezultatul este scris in fisier in ordine inversa.

Varianta 1 – Send, Recv

Se citeste dimensiunea fiecarui numar si se determina care este mai mare dintre cele doua. In procesul 0 se citesc numerele iar daca un numar este mai mare decat celalalt se adauga zerouri. Se impart datele egal la numarul de procese. Unele procese pot avea un element in plus in urma impartirii din cauza restului. Datele sunt trimise din procesul 0 catre celelalte procese unde se aduna numerele cifra cu cifra. Daca apare carry acesta este trimis catre urmatorul process. Primul process nu primeste carry, iar ultimul process nu trimite carry ci adauga 1 pe ultima pozitie. Rezultatul este scris in fisier in procesul 0.

Varianta 2 – Scatter

Se citeste dimensiunea fiecarui numar si se determina care este mai mare dintre cele doua. Deoarece dimensiunea trebuie sa se imparta la numarul de procese se calculeaza o noua dimensiune care se imparte la numarul de procese si este mai mare sau egala cu cea curenta. Se adauga zerouri pentru elementele adaugate in plus. In procesul 0 se citesc numerele iar daca un numar este mai mare decat celalalt se adauga zerouri. Datele sunt impartite proceselor folosind Scatter. Procesele fac suma cifrelor si in cazult in care apare carry este trimis la umratorul process. Rezultatul final este obtinut in procesul 0 cu Gather, iar in cazul in care apare carry se adauga 1 pe ultima pozitie. Rezultatul este scris in fisier in procesul 0.

Date intrare	Procese	Timp executie
N1 = N2 = 123456789123456789	secvential	0.598
	MPI Send Recv -n 4	1.7
	MPI Send Recv -n 8	3.299
	MPI Scatter -n 4	1.8
	MPI Scatter -n 8	2.999
sizeN1 = sizeN2 = 1000	secvential	2.498
	MPI Send Recv -n 4	2.198
	MPI Send Recv -n 8	3.699
	MPI Scatter -n 4	2.498
	MPI Scatter -n 8	3.7
sizeN1 = 100 sizeN2 = 100000	secvential	21.501
	MPI Send Recv -n 4	23.110
	MPI Send Recv -n 8	27.603
	MPI Scatter -n 4	23.7
	MPI Scatter -n 8	26.604

Analiza

- In majoritatea cazurilor timpul de executie al algoritmului secvential a fost apropiat de cel cu MPI.
- Timpul de executie este mereu mai mic in cazul secvential, mai putin in cazul in care dimensiunea numerelor este 1000 si numarul de procese este 4 utilizand Send Recv.
- Timpul de executie este mereu mai mare in cazul in care numarul proceselor este 8 decat in cazul in care numarul proceselor este 4.
- Intre Send Recv si Scatter nu este o diferenta mare. In general Send Recv este mai rapid.



